AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

1. (currently amended) A process for monitoring, comprising:

accessing a method;

automatically determining whether to modify said method, said step of automatically determining whether to modify said method includes automatically determining whether said method calls another method and whether said method has an access level that satisfies a criterion; and

modifying said method for a particular purpose only if said method calls another method and said access level satisfies said criterion.

2. (currently amended) The process according to claim 1, wherein:

said ~~step of~~ determining whether to modify said method includes automatically determining whether said method is a synthetic method; and

said ~~step of~~ modifying includes modifying said method only if said method is not a synthetic method, said access level satisfies said criterion and said method calls another method.

3-4. (cancelled)

5. (currently amended) The process according to claim 1, wherein:

said ~~step of~~ determining whether to modify said method includes automatically determining whether said method is a synthetic method and has an access level of public or package in the JAVA programming language; and

said ~~step of~~ modifying includes modifying said method only if said method is not a synthetic method, has said access level of public or package, and said method calls another method.

6. (currently amended) The process according to claim 1, wherein:

said ~~step of~~ determining whether said method has an access level that satisfies a criterion ~~to~~

~~modify said method~~ includes automatically determining whether said method can be called by a sufficient scope of one or more other methods~~, said determining whether said method can be called by a sufficient scope of one or more other methods is based on an access level of said method~~; and

said ~~step of~~ modifying said method includes modifying said method only if said method can be called by said sufficient scope of one or more other methods and said method calls another method.

7. (currently amended) The process according to claim 1, wherein:

said ~~step of~~ modifying includes modifying object code.

8. (currently amended) The process according to claim 1, wherein:

said ~~step of~~ modifying includes adding a tracer for said method.

9. (currently amended) The process according to claim 1, wherein:

said ~~step of~~ modifying includes adding a timer for said method.

10. (currently amended) The process according to claim 1, wherein:

said ~~step of~~ modifying includes adding exit code and start code to existing object code.

11. (currently amended) The process according to claim 10, wherein:

said start code starts a tracing process;

said exit code stops said tracing process;

said exit code is positioned to be executed subsequent to original object code;

said ~~step of~~ adding exit code includes adding an instruction to jump to said exit code from said original object code;

said ~~step of~~ adding exit code includes adding an entry in an exceptions table; and

said ~~step of~~ adding an entry in said exceptions table includes adding a new entry into said exceptions table for said method, said new entry indicates a range of indices corresponding to said original object code, said new entry includes a reference to said exit code and said new entry

indicates that said new entry pertains to all types of exceptions.

12. (previously presented) The process according to claim 1, wherein:

said particular purpose is to add a first tracer.

13. (currently amended) A process for monitoring, comprising:

automatically determining which methods of a set of methods call one or more other methods[[,]] and are synthetic; and

using a first tracing mechanism for said methods that call one or more other methods and are not synthetic without using said first tracing mechanism for methods that do not call one or more other methods or are synthetic.

14-16. (cancelled)

17. (currently amended) The process according to claim 13, wherein:

said ~~step of~~ automatically determining includes automatically determining whether said methods have an access level of public or package in the JAVA programming language; and

said ~~step of~~ using includes using said first tracing mechanism only if said methods are not synthetic methods, have said access level of public or package in the JAVA programming language, and said methods call one or more other methods.

18. (currently amended) The process according to claim 13, wherein:

said ~~step of~~ determining includes automatically determining whether said methods can be called by a sufficient scope of one or more other methods; and

said ~~step of~~ using includes using said first tracing mechanism only if said methods can be called by said sufficient scope of one or more other methods, said methods are not synthetic, and said methods call one or more other methods.

19. (currently amended) The process according to claim 13, wherein:

said ~~step of~~ using a first tracing mechanism includes adding and using timers for said methods.

20. (currently amended) The process according to claim 13, wherein:

said ~~step of~~ using a first tracing mechanism includes modifying existing object code to add said first tracing mechanism.

21. (previously presented) The process according to claim 20, wherein:

said first tracing mechanism includes timers for said methods.

22. (previously presented) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a process comprising:

automatically determining which methods of a set of methods to modify, said step of determining includes automatically determining which methods call one or more other methods and have an access level of either public or package in the JAVA programming language; and

modifying for a particular purpose only those methods that call one or more other methods and have an access level of either public or package in the JAVA programming language.

23. (currently amended) The one or more processor readable storage devices according to claim 22, wherein:

said ~~step of~~ automatically determining includes automatically determining whether said methods are not synthetic methods; and

said ~~step of~~ modifying includes modifying said methods only if said methods are determined to not be synthetic methods, said methods have an access level of either public or package in the JAVA programming language and said methods call one or more other methods.

24-27. (cancelled)

28. (currently amended) The one or more processor readable storage devices according to claim 22, wherein:

said ~~step of~~ modifying includes modifying existing object code.

29. (currently amended) The one or more processor readable storage devices according to claim 22, wherein:

said ~~step of~~ modifying includes adding tracers.

30. (currently amended) The one or more processor readable storage devices according to claim 22, wherein:

said ~~step of~~ modifying includes adding timers.

31. (cancelled)

32. (currently amended) The one or more processor readable storage devices according to claim 22, wherein:

said start code starts a tracing process;

said exit code stops said tracing process;

said exit code is positioned to be executed subsequent to original object code;

said step of adding exit code includes adding an instruction to jump to said exit code from said original object code;

said ~~step of~~ adding exit code includes adding an entry in an exceptions table; and

said ~~step of~~ adding an entry in said exceptions table includes adding a new entry into said exceptions table for said method, said new entry indicates a range of indices corresponding to said original object code, said new entry includes a reference to said exit code and said new entry indicates that said new entry pertains to all types of exceptions.

33. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor

readable code for programming one or more processors to perform a process comprising:

automatically determining whether to trace a method, said step of determining includes automatically determining whether said method calls another method and if said method has an access level that satisfies a criterion; and

tracing said method for a particular purpose only if said method calls another method and said access level satisfies the criterion.

34. (currently amended) The one or more processor readable storage devices according to claim 33, wherein:

said ~~step of~~ determining includes automatically determining whether or not said method is flagged by a compiler as being synthetic; and

said ~~step of~~ tracing includes tracing said method only if said method is not flagged by said compiler as being synthetic and said method calls another method.

35. (currently amended) The one or more processor readable storage devices according to claim 33, wherein:

said ~~step of~~ automatically determining includes automatically determining whether said method has an access level of public or package in the JAVA programming language, an access level of public indicates that a method can be called by a method in a class of any parentage, an access level of package indicates that a method can be called by methods in classes in the same package regardless of parentage; and

said ~~step of~~ tracing includes tracing said method only if said method is determined to have said access level of public or package and said method calls another method.

36. (cancelled)

37. (currently amended) The one or more processor readable storage devices according to claim 33, wherein:

said ~~step of~~ automatically determining includes automatically determining whether said

method is not a synthetic method and has an access level of public or package, said access level is one of a plurality of access levels in a JAVA programming language; and

said ~~step of~~ tracing includes tracing said method only if said method is determined to not be a synthetic method, have said access level of public or package, and said method calls another method.

38. (currently amended) The one or more processor readable storage devices according to claim 33, wherein:

said ~~step of~~ tracing includes timing said method.

39. (previously presented) An apparatus capable of monitoring, comprising:

means for automatically determining whether a method calls another method;

means for automatically determining whether said method can be called by a sufficient scope of one or more other methods;

means for automatically determining whether said method is not a synthetic method; and

means for tracing said method for a particular purpose only if said method calls another method, said method can be called by a sufficient scope of one or more other methods, and said method is not a synthetic method.

40. (previously presented) An apparatus capable of monitoring, comprising:

a storage device; and

one or more processors in communication with said storage device, said one or more processors perform a process comprising:

accessing a method,

determining whether said method calls one or more different methods and can be called by a sufficient scope of one or more other methods, and

tracing said method for a particular purpose only if said method calls one or more different methods and can be called by a sufficient scope of one or more other methods.

41. (currently amended) The apparatus according to claim 40, wherein:

said ~~step of~~ determining includes determining whether said method is not a synthetic method; and

said ~~step of~~ tracing includes tracing said method only if said method is determined to not be a synthetic method and said method calls one or more different methods.

42. (currently amended) The apparatus according to claim 40, wherein:

said ~~step of~~ determining includes determining whether said method has an access level of public or package in the JAVA programming language; and

said ~~step of~~ tracing includes tracing said method only if said method is determined to have said access level of public or package and said method calls one or more different methods.

43. (cancelled)

44. (previously presented) The apparatus according to claim 40, wherein:

said process further includes modifying existing object code for said method in order to add a first tracing mechanism.

45. (previously presented) The apparatus according to claim 44, wherein:

said first tracing mechanism includes a timer.

46. (currently amended) The apparatus according to claim 40, wherein:

said ~~step of~~ tracing includes timing said method.

47. (previously presented) A process for monitoring, comprising:

accessing a method;

automatically determining whether said method is complex, said step of automatically determining includes automatically determining that said method is complex if said method satisfies the following criteria:

said method calls another method;

said method has an access level of public or package in the JAVA programming language; and

said method is not flagged by a compiler as being synthetic; and

adding a tracer to said method only if said method is automatically determined to be complex.

48-50. (cancelled)

51. (currently amended) The process according to claim 5, wherein:

said ~~step of~~ modifying includes adding a tracer for said method.

52. (currently amended) The apparatus according to claim 40, wherein:

said ~~step of~~ determining includes determining whether said method is not a synthetic method and whether said method has an access level of public or package in the JAVA programming language; and

said ~~step of~~ tracing includes tracing said method only if said method is determined to not be a synthetic method, said method is determined to have an access level of public or package, and said method calls one or more different methods.

53-59. (cancelled)